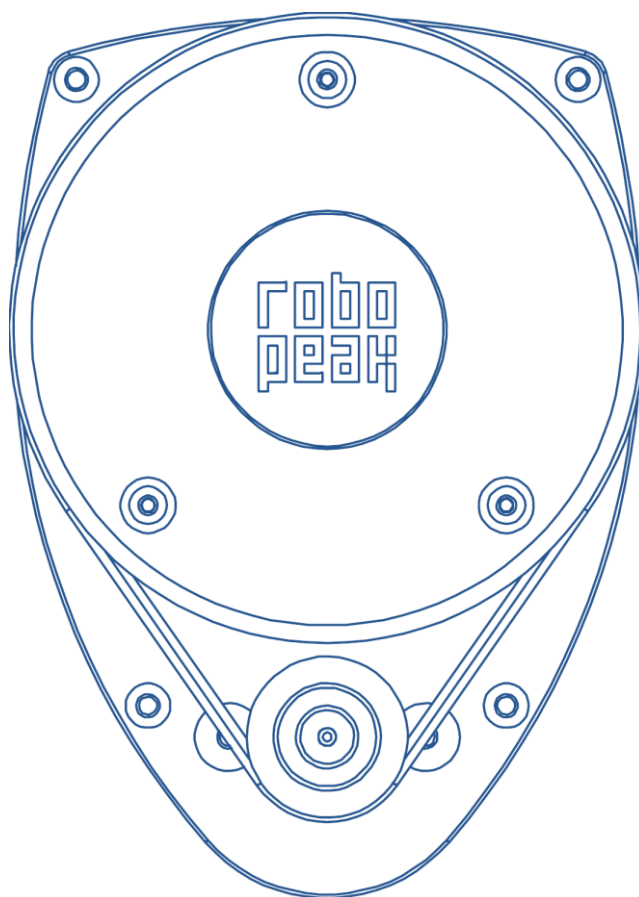




## RPLIDAR 应用笔记

Arduino 驱动支持及实例程序

2014-5  
Rev.1



## 目录:

---

<b>1. 简介.....</b>	<b>2</b>
性能问题.....	2
<b>2. RPLIDAR ARDUINO 驱动库 .....</b>	<b>3</b>
关于该驱动库.....	3
系统需求.....	3
驱动库的安装.....	3
示例程序.....	4
<b>3. 示例程序的介绍和使用.....</b>	<b>6</b>
基本介绍.....	6
例子: SIMPLE_CONNECT.....	8
例子: DISTANCE_TO_COLOR.....	9
<b>4. 修订历史 .....</b>	<b>11</b>

## 1. 简介

---

为了方便用户可以快速的在 Arduino 系统中使用 RPLIDAR, RoboPeak 提供了为 Arduino 设计的 RPLIDAR 驱动库以及配套的例子程序。

该 Arduino 驱动库可以在标准的基于 Atmel AVR 芯片的 Arduino 开发板, 比如 Arduino UNO 上执行, 也可以支持其他 Arduino 兼容板, 例如 Intel Galileo、Leaf Lab Maple 等。

该驱动库提供的 API 与标准 RPLIDAR SDK 中所定义的非常相似。

本文档也会介绍几个配套的实例程序。

### 性能问题

在使用本 **RPLIDAR Arduino** 驱动库时, 用户需要十分注意所编写代码的执行效率问题。

由于 RPLIDAR 在工作时将以每秒 2000 个采样点的数据量向宿主系统持续的输出, 因此作为宿主系统必须在 **500us** 以内完成一次采样点的数据处理。

对于诸如 Arduino UNO、Arduino Mega 这类标准的基于 Atmel AVR 芯片的 Arduino 板而言, 其相对低速的 MCU 就更加要求应用程序必须足够的高效以便能在有限的时间中完成对一次采样数据点的处理。不然的话, 就会出现数据的丢失。

而对于诸如 SLAM 这类复杂的应用而言, RoboPeak 推荐用户使用更强大的硬件平台来完成, 而不是标准 Arduino 系统。

## 2. RPLIDAR Arduino 驱动库

### 关于该驱动库

RPLIDAR 的 Arduino 驱动库可以从 RPLIDAR 的产品网站中获取:

<http://rplidar.robopeak.com>

其源代码可以从 github 中获取:

[https://github.com/robopeak/rplidar\\_arduino](https://github.com/robopeak/rplidar_arduino)

### 系统需求

- Arduino IDE:  
1.0.5 或以上版本
- 硬件:  
Arduino 板或者至少具备一个硬件串口的 Arduino 兼容硬件  
型号为 A1M1 的 RPLIDAR 开发套装

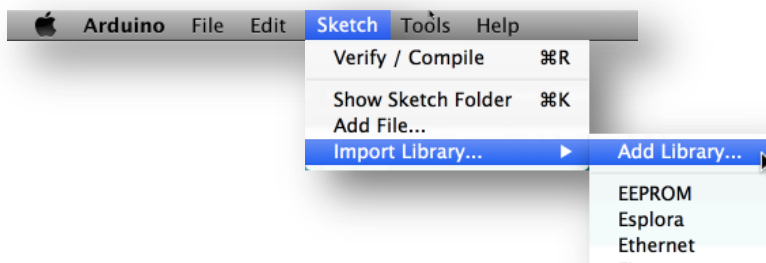
### 驱动库的安装

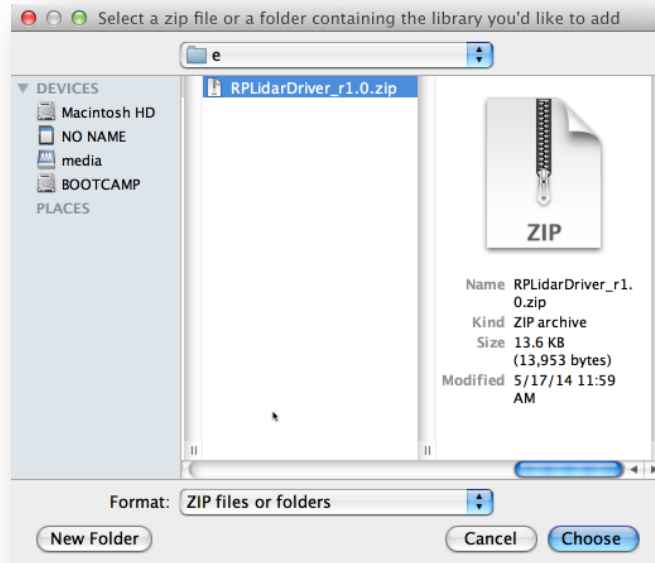
该驱动库是以标准的 Arduino 库的形式提供的, 即可以在 Arduino IDE 中使用导入库命令安装的 ZIP 压缩包。

请按照如下步骤完成安装, 或者可以参考 Arduino 官网了解详情:

<http://arduino.cc/en/Guide/Libraries>

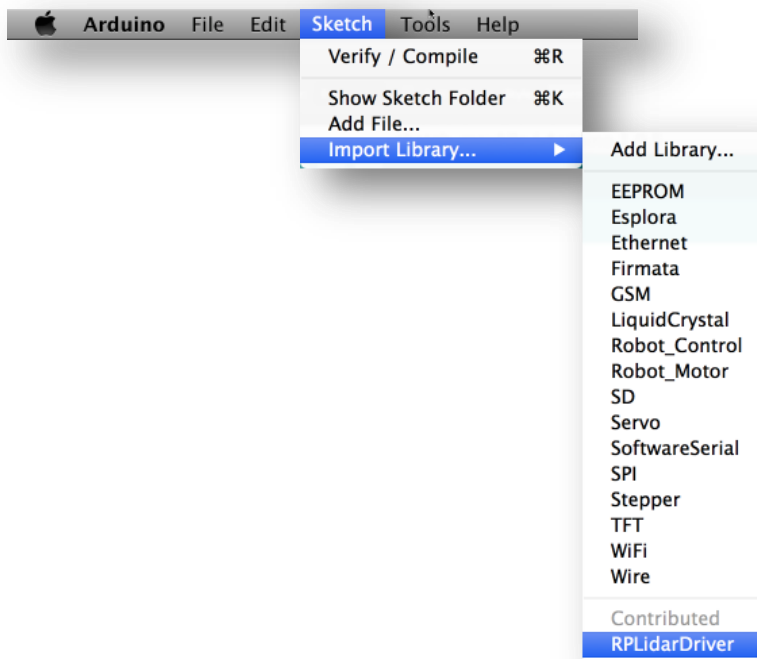
- 步骤#1  
下载 Arduino 驱动的 ZIP 包, 并使用 Arduino IDE 的增加库菜单命令:





- 步骤#2

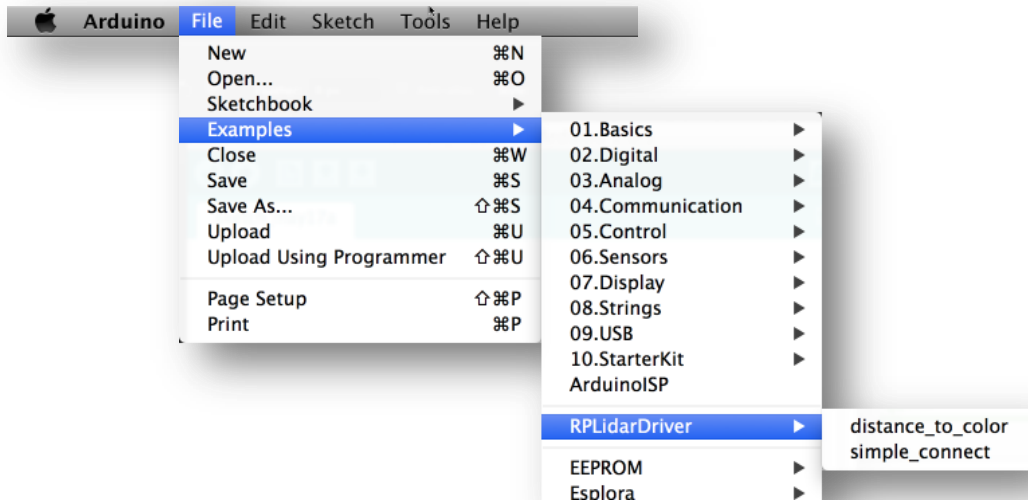
当驱动库增加成功后，您可以在可用库列表菜单中看到本驱动库的条目。



## 示例程序

您可以通过参考驱动库附带的例子代码开始使用本驱动库。

当驱动库安装完毕后，您可以在例子列表菜单中看到他们：



```

simple_connect | Arduino 1.0.5

simple_connect

// This sketch code is based on the RPLIDAR driver library provided by RoboPeak
#include <RPLidar.h>

// You need to create an driver instance
RPLidar lidar;

#define RPLIDAR_MOTOR 3 // The PWM pin for control the speed of RPLIDAR's motor.
                       // This pin should connected with the RPLIDAR's MOTOCCTRL signal

void setup() {
  // bind the RPLIDAR driver to the arduino hardware serial
  lidar.begin(Serial);

  // set pin modes
  pinMode(RPLIDAR_MOTOR, OUTPUT);
}

void loop() {
  if (IS_OK(lidar.waitPoint())) {
    float distance = lidar.getCurrentPoint().distance; //distance value in mm unit
    float angle = lidar.getCurrentPoint().angle; //angle value in degree
    bool startBit = lidar.getCurrentPoint().startBit; //whether this point is belong to a new scan
    byte quality = lidar.getCurrentPoint().quality; //quality of the current measurement

    //perform data processing here...

  } else {
    analogWrite(RPLIDAR_MOTOR, 0); //stop the rplidar motor

    // try to detect RPLIDAR...
    rplidar_response_device_info_t info;
    if (IS_OK(lidar.getDeviceInfo(info, 100))) {

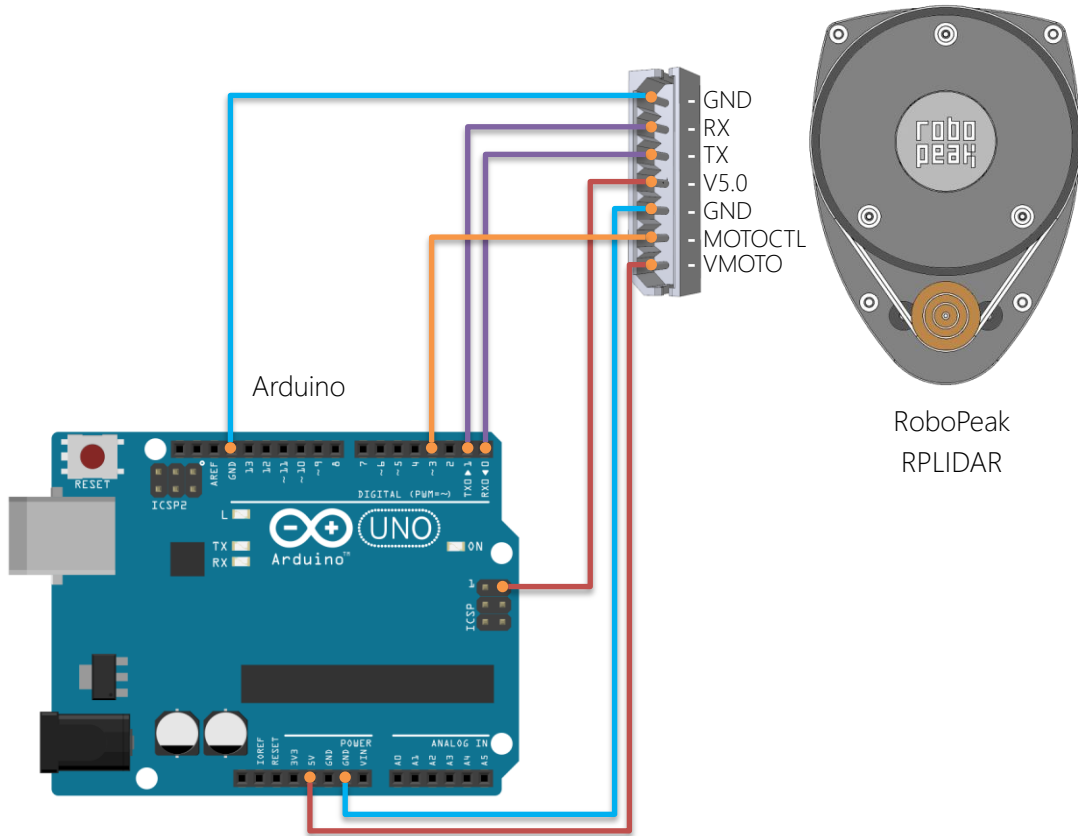
```

### 3. 示例程序的介绍和使用

#### 基本介绍

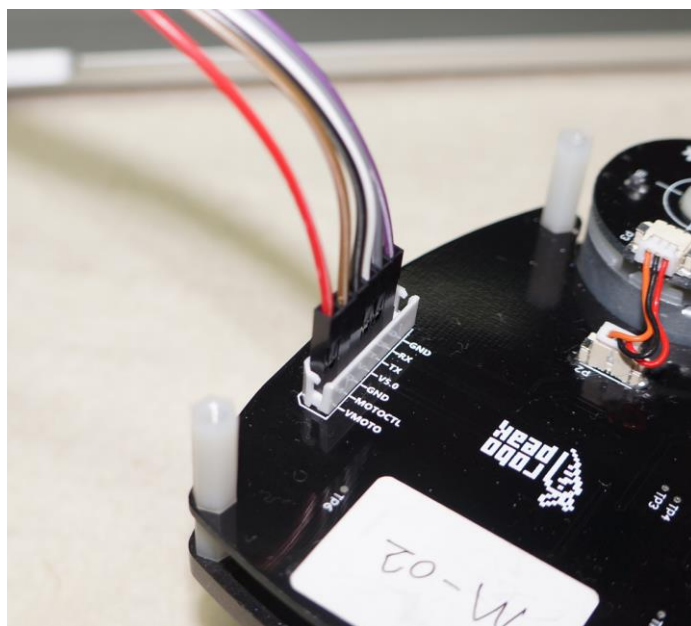
- 系统连接

按照下图所示，将 RPLIDAR 与 Arduino 板相连。可以使用 2.5mm 规范的杜邦线完成连接。



RPLIDAR 引脚	Arduino 引脚	描述
GND	GND	电源地线
V5.0	5V	给 RPLIDAR Core 供电
VMOTO	5V/ICSP Pin2	给 RPLIDAR 电机供电
RX	Pin1 (TXD)	串口, RX<-TX
TX	Pin0 (RXD)	串口, TX->RX
MOTOCTL	Pin3 (PWM)	RPLIDAR 电机调速控制

如果您只希望 RPLIDAR 电机以最高速度运转，则可以将 MOTOCTL 引脚连接在 Arduino 板的 3V3 引脚上。



- 在程序下载时的注意事项

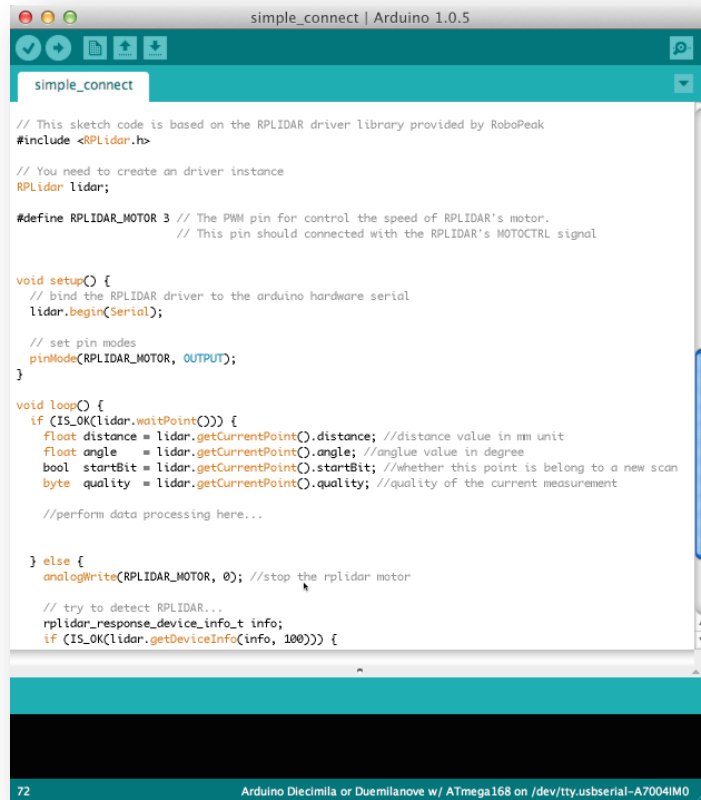
由于 RXD, TXD 在 Arduino 下载程序时会被占用, 因此在开始下载程序前, 需要将他们与 RPLIDAR 断开。不然的话, 下载程序过程会失败。



## 例子: simple\_connect

该例子展示了连接 RPLIDAR，开始测距扫描并获取采样数据的一般过程。您可以使用该例子作为今后应用的模版。

`waitPoint()`方法会在 Arduino 接收到一个采样数据点后返回 `RESULT_OK`。用户可以在该代码段内增加必要的处理逻辑。



```
simple_connect | Arduino 1.0.5

// This sketch code is based on the RPLIDAR driver library provided by RoboPeak
#include <RPLidar.h>

// You need to create an driver instance
RPLidar lidar;

#define RPLIDAR_MOTOR 3 // The PWM pin for control the speed of RPLIDAR's motor.
                       // This pin should connected with the RPLIDAR's MOTOCTRL signal

void setup() {
  // bind the RPLIDAR driver to the arduino hardware serial
  lidar.begin(Serial);

  // set pin modes
  pinMode(RPLIDAR_MOTOR, OUTPUT);
}

void loop() {
  if (IS_OK(lidar.waitPoint())) {
    float distance = lidar.getCurrentPoint().distance; //distance value in mm unit
    float angle = lidar.getCurrentPoint().angle; //angle value in degree
    bool startBit = lidar.getCurrentPoint().startBit; //whether this point is belong to a new scan
    byte quality = lidar.getCurrentPoint().quality; //quality of the current measurement

    //perform data processing here...

  } else {
    analogWrite(RPLIDAR_MOTOR, 0); //stop the rplidar motor

    // try to detect RPLIDAR...
    rplidar_response_device_info_t info;
    if (IS_OK(lidar.getDeviceInfo(info, 100))) {

```

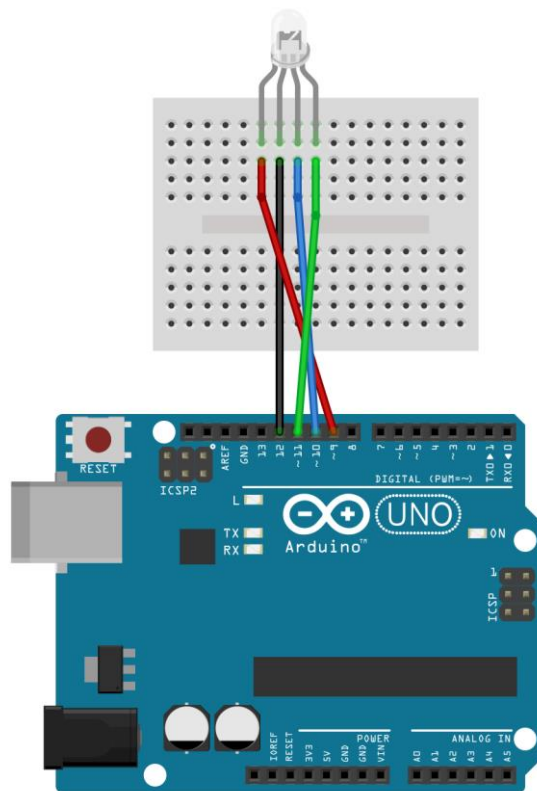
## 例子: distance\_to\_color

该例子使用了一个 RGB 彩色 LED 用于显示被 RPLIDAR 检测到的最近物体的方位角以及距离。

物体的方位角度通过 LED 的色彩来表现。而物体的距离通过 LED 的明暗程度来表现。

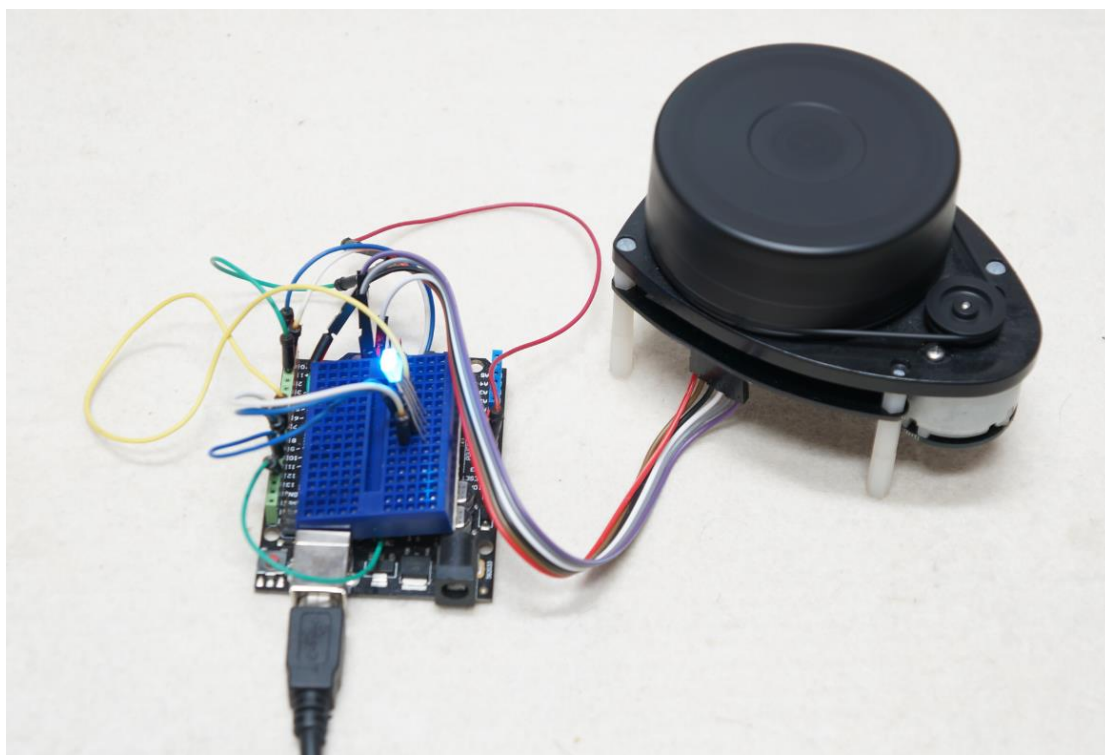
- 系统连接

请先按照前面基本介绍章节的步骤完成连接，随后做出如下额外的连接：



LED 引脚	Arduino 引脚
共阳极	12
红	9
绿	11
蓝	10

完整的系统连接如下所示：



您可以参考如下视频了解本例子的具体效果：

<http://www.tudou.com/programs/view/rf2Lo7ov95U/>

## 4. 修订历史

---

日期	内容
2013-5-17	第一稿